



Centre de recherche sur les transports (C.R.T.)
Centre for Research on Transportation

CRT-2003-38

**EFFICIENT NEIGHBORHOOD SEARCH FOR THE
PROBABILISTIC PICKUP AND DELIVERY
TRAVELLING SALESMAN PROBLEM**

by

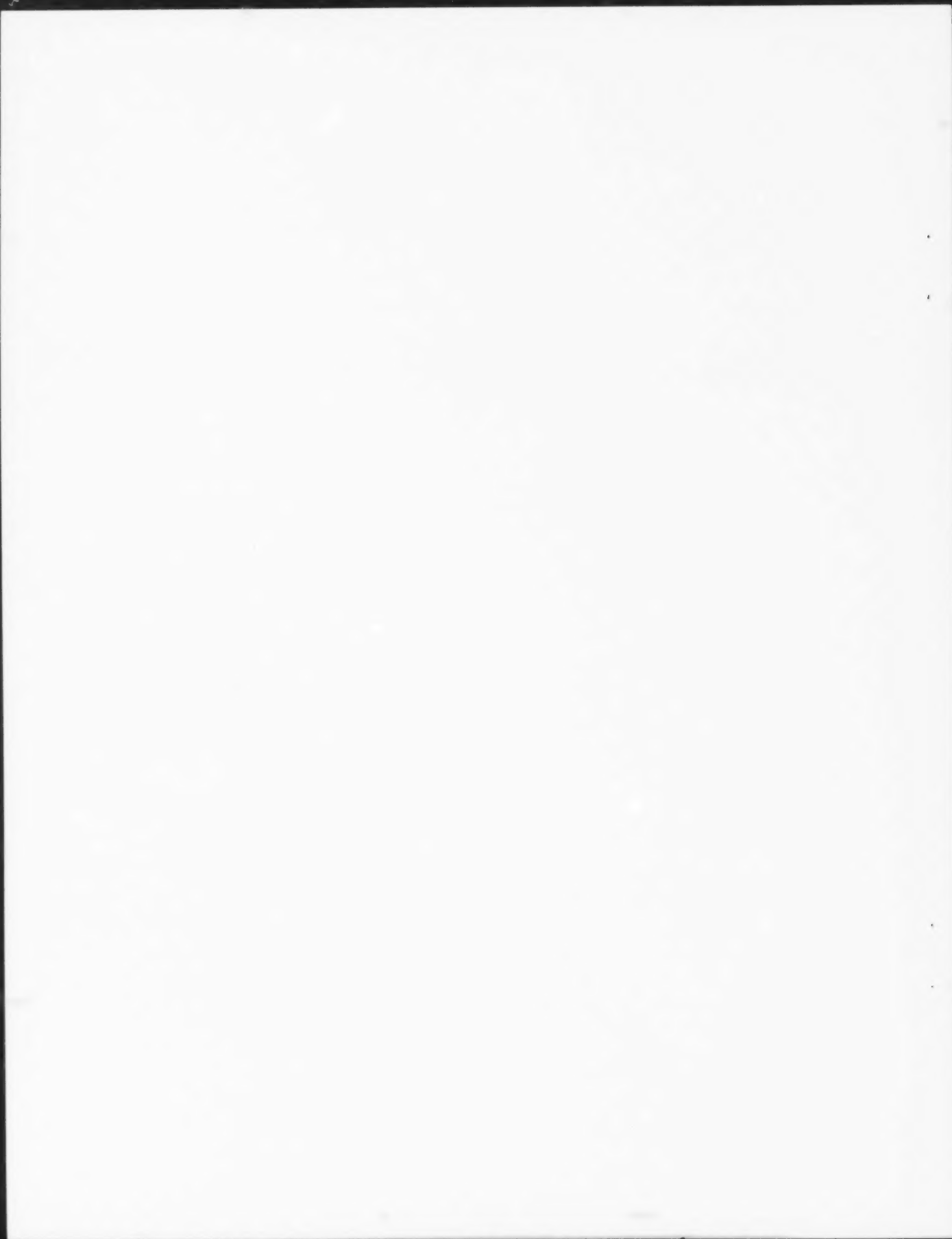
**Patrizia Beraldi
Gianpaolo Ghiani
Gilbert Laporte
Roberto Musmanno**

December 2003

Université de Montréal
C.P. 6128, succursale Centre-ville, Montréal QC H3C 3J7 Canada
Téléphone: (514) 343-7575 / Télécopieur: (514) 343-7121

crt@crt.umontreal.ca
www.crt.umontreal.ca





**PUBLICATION
CRT-2003-38**

**EFFICIENT NEIGHBORHOOD SEARCH FOR THE
PROBABILISTIC PICKUP AND DELIVERY
TRAVELLING SALESMAN PROBLEM**

by

**Patrizia Beraldi¹
Gianpaolo Ghiani²
Gilbert Laporte³
Roberto Musmanno¹**

This research was supported in part by the Italian National Research Council, by the Center of Excellence for High-Performance Computing, University of Calabria, Italy, and by the Natural Sciences and Engineering Research Council of Canada (NSERC).

- ¹. Centro di Eccellenza sul Calcolo ad Alte Prestazioni, Università degli Studi della Calabria, 87030 Rende (CS), Italy (beraldi, musmanno@unical.it)
- ². Dipartimento di Ingegneria dell'Innovazione, Università degli Studi di Lecce, 73100 Lecce, Italy (gianpaolo.ghiani@unile.it)
- ³. Canada Research Chair in Distribution Management, HEC Montréal, 3000 Côte-Sainte-Catherine, Montréal, Canada H3T 2A7, and Centre de recherche sur les transports, Université de Montréal, C.P. 6128, succursale Centre-ville, Montréal, Canada H3C 3J7 (gilbert@crt.umontreal.ca)

Dépôt légal – Bibliothèque nationale du Québec
Bibliothèque nationale du Canada, 2003

Abstract

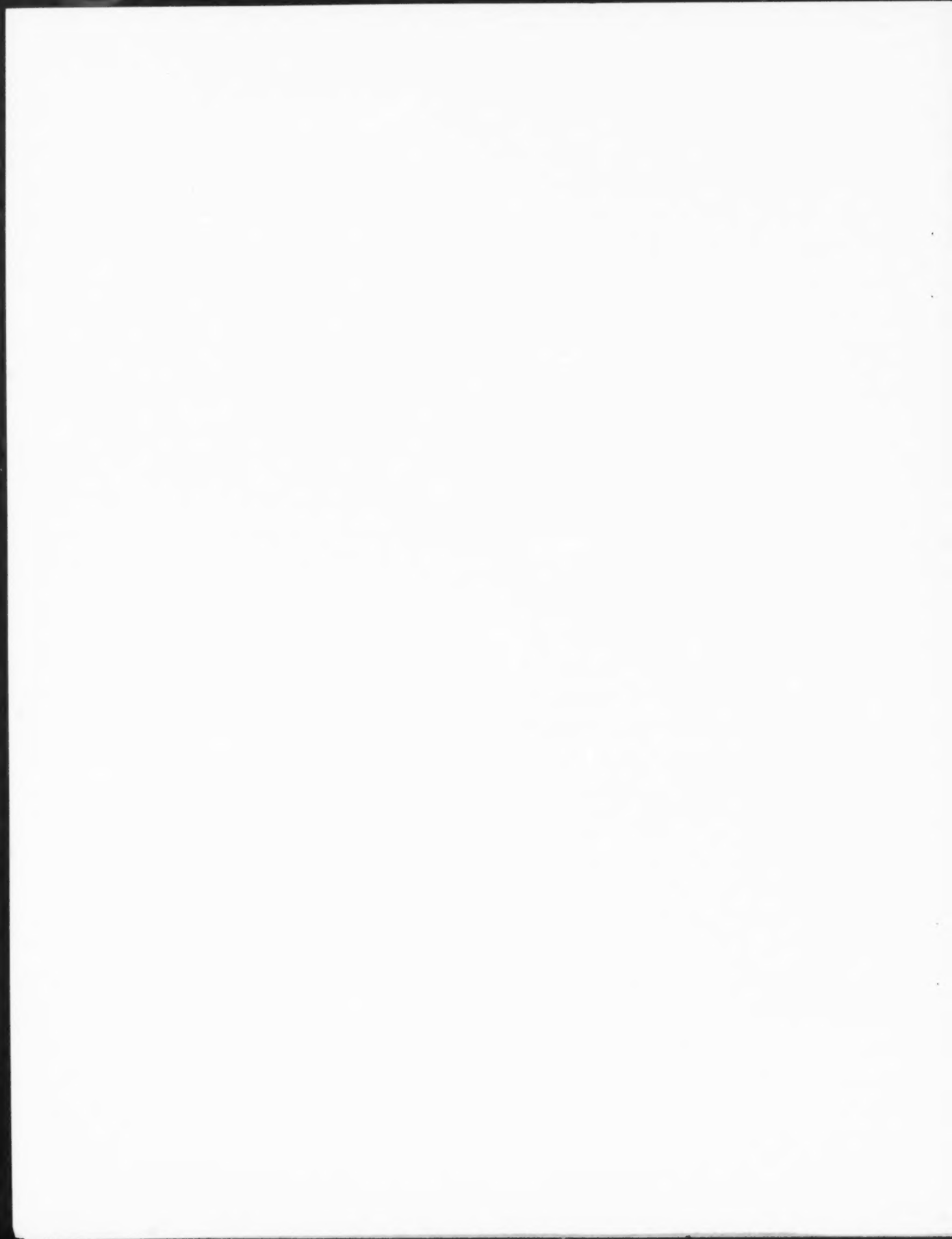
In this paper the probabilistic Pick-up and Delivery Travelling Salesman Problem is studied and an efficient neighborhood search procedure is developed. This procedure requires an $O(n^3)$ of computations for the evaluation of the neighborhood of a given solution while the straightforward approach has a $O(n^5)$ complexity. Computational results show that, if embedded in a local search or Tabu Search framework, our neighborhood evaluation technique results in an impressive computation time reduction.

Keywords: probabilistic pick-up and delivery travelling salesman problem, tabu search.

Résumé

Dans cet article, on étudie le problème probabiliste du voyageur de commerce avec cueillettes et livraisons et on développe une procédure de recherche par voisinages efficiente pour ce problème. La procédure requiert $O(n^3)$ opérations pour l'évaluation du voisinage d'une solution donnée tandis qu'une approche directe requiert $O(n^5)$ opérations. Des résultats numériques montrent que si notre technique d'évaluation est imbriquée dans une procédure de recherche locale ou dans une méthode de recherche avec tabous, la réduction des temps de calculs est significative.

Mots-clés : problème probabiliste du voyageur de commerce avec cueillettes et livraisons, recherche avec tabous.



1 Introduction

This paper deals with the probabilistic version of the Pick-up and Delivery Travelling Salesman Problem (PDTSP). In its deterministic version, the PDTSP can be defined as follows. Let $G = (V, A)$ be a complete graph where $V = \{0, 1, \dots, n, n+1, \dots, 2n\}$ is the vertex set and A is the arc set. Vertex 0 denotes the depot where a single vehicle is located, whereas vertices i and $i+n$ represent the pick-up and delivery points of the i -th transportation request, respectively ($i = 1, \dots, n$). In what follows, we refer $i+n$ to as the *companion* of i , and vice versa. With each arc (i, j) is associated a nonnegative travel cost c_{ij} . Travel costs are assumed to satisfy the triangle inequality.

The PDTSP consists of designing a tour $\tau = (i_0, i_1, i_2, \dots, i_{2n}, i_{2n+1})$ such that:

- (i) $i_0 = i_{2n+1} = 0$;
- (ii) $\{i_1, \dots, i_{2n}\} = V \setminus \{0\}$;
- (iii) vertex i is visited before vertex $i+n$;
- (iv) the total cost is minimized.

In this paper we study the probabilistic version of the PDTSP (PPDTSP) in which each vertex i is associated to a Bernoulli random variable ξ_i , taking value 1 if i requires service, and 0 otherwise. Let p_i denote the probability that vertex i must be served. Vertices with $p_i = 1$ are referred to *black* vertices, while vertices with $0 < p_i < 1$ are called *white* vertices. As customary in two stage stochastic programming with recourse [1], two decisions have to be made. In a first stage we have to determine an a priori feasible tour over G . In the second stage, requested vertices are visited in the same order in which they appear in the a priori tour, whereas unrequested vertices are simply skipped. The objective is to find an a priori tour whose *expected cost* is minimum. Applications of the PPDTSP arise in the design of dial-a-ride transportation systems as well as of freight pickup and delivery systems.

The PPDTSP is a stochastic vehicle routing problem [11] which has not been addressed explicitly so far. However, there are few papers which are related to our problem. Swihart and Papastavrou [31] analyze several policies for a

single vehicle stochastic and dynamic pick-up and delivery problem in which demands for service arrive according to a Poisson process in time. The pick-up locations of the demands are independent and uniformly distributed over a service region. The objective is to minimize the expected time in the system for the demands. Upper and lower bounds on the performance of the routing policies are derived for light and heavy traffic. Jaillet [14] derive an $O(n^3)$ formula for the expected cost of an a priori solution of the probabilistic Travelling Salesman Problem (TSP). Laporte, Louveaux and Mercure [17] describe an exact algorithm for determining an a priori TSP tour of least expected cost. The deterministic counterpart of our problem has been examined in a few papers. Kalantari et al. [16] have described an exact branch-and-bound procedure for this problem which has been tested on instances involving no more than 40 vertices. Ruland and Rodin [25] have developed four families of PDTSP valid inequalities which have been embedded into a branch-and-cut algorithm. Savelsbergh [27] and Healy and Moll [12] have proposed several heuristic improvement schemes based on edge interchanges. Renaud, Boctor and Ouenniche [23] have implemented an edge-interchange scheme as well as a vertex deletion and reinsertion method. More recently, Renaud, Boctor and Laporte [22] have described and compared seven perturbation heuristics which consistently produce high quality solutions.

The PDTSP is related to, but different from, the dial-a-ride problem (DARP) in the following way: in the DARP there may be several capacitated vehicles instead of only one, time windows are often present, and the objective function may be different. For example, one may wish to minimize the total distance travelled between each pickup point and its associated delivery point. For references on the DARP, see, e.g., Psaraftis [19] [20] [21], Sexton and Bodin [28] [29], Sexton and Choi [30], Desrosiers et al. [5], Jaw et al. [15], Dumas et al. [6], Van der Bruggen et al. [32], Savelsbergh and Sol [26], Hunsaker and Savelsberg [13], and Cordeau and Laporte [3] [4]. Two other problems related to the PDTSP are the vehicle routing problem with backhauls [2] [10] [18], and the TSP with Backhauls [7] [8]. In both problems, all pickup customers must be visited before delivery customers but there is no relationship between these two sets.

Not only the PPDTSP is NP-hard, but also generating high quality heuristic solutions can be time consuming. In this paper we illustrate a neighborhood evaluation procedure which is two order of magnitude faster than the straightforward approach. The remainder of this paper is organized as follows. In Section 2 the expected cost of an a priori PPDTSP solution is computed while in section 3 an $O(n^3)$ neighborhood evaluation procedure is illustrated. Computational results reported in Section 4 show that such a technique allows achieving remarkable computation time reduction when compared to the straightforward $O(n^5)$ neighborhood evaluation procedure.

2 Expected cost of an a priori solution

In order to compute the expected cost of a *given* a priori solution, we should consider all possible combinations of the realizations of random variables ξ_i , thus requiring an exponential number of operations. Fortunately, we can reduce the complexity of computing the expected length to $O(n^3)$, by using arguments similar to Jaillet [14].

Let L_τ be the length of a tour τ and $E[L_\tau]$ its expected value. Given two vertices $i_u, i_v \in V \setminus \{0\}$, $u < v$, let $I_1^{u,v} = \{u\}$ if $i_v = i_u + n$, or $I_1^{u,v} = \{u, v\}$ otherwise, $I_2^{u,v} = \{r \in \{u+1, \dots, v-1\} : i_r \leq n\}$ and $I_3^{u,v} = \{r \in \{u+1, \dots, v-1\} : i_r \geq n+1, i_r = i_{r'} + n, r' \in \{1, \dots, u-1\}\}$. The expected cost of τ can be determined as follows.

Proposition 2.1 *Given an a priori feasible tour τ , the probability $p_\tau(i_u, i_v)$ that arc (i_u, i_v) is traversed is equal to 0 if $u > v$. Otherwise, it is equal to 0 if there exists a vertex $h \in \{u+1, \dots, v-1\}$ such that $i_v = i_h + n$, 0 if there exists a vertex $k \in \{u+1, \dots, v-1\}$ such that $i_k = i_u + n$, or*

$$\prod_{h \in I_1^{u,v}} p_{i_h} \prod_{k \in I_2^{u,v}} (1 - p_{i_k}) \prod_{w \in I_3^{u,v}} (1 - p_{i_w}) \quad (1)$$

otherwise.

Proof. If $u > v$ no recourse action will cause arc (i_u, i_v) to be traversed. If $u < v$, the probability to traverse arc (i_u, i_v) is zero if chain $i_{u+1}, \dots, i_{v-1}$ contains the companion of i_u or i_v . Finally, if chain $i_{u+1}, \dots, i_{v-1}$ contains both the pick-up and delivery nodes of a subset of requests, (1) holds.

Hence, the expected length $E[L_\tau]$ of the a priori tour τ is given by:

$$E[L_\tau] = \sum_{u \in \{0,1,\dots,2n\}, v \in \{u+1,\dots,2n+1\}} p_\tau(i_u, i_v) c_{i_u i_v} \quad (2)$$

3 Neighborhood evaluation

Let $\tau = (i_0, i_1, i_2, \dots, i_{2n}, i_{2n+1})$ be the current a priori tour. τ can be feasible or not depending on whether condition (iii) is satisfied or not. A neighbor $\tau' = (i_0, i_1, \dots, i_{s-1}, i_{s+1}, \dots, i_t, i_s, \dots, i_{n+1})$ of τ can be obtained through a *move* which eliminates a vertex i_s from τ and re-inserts it in the t -th position ($s \neq t$). Let $N(\tau)$ be the resulting neighborhood.

Proposition 3.1 *The determination of the least cost neighbor of a given tour τ takes $O(n^5)$ computations if each neighbor is evaluated from scratch by means of formula (2).*

Proof. *Each of the $O(n)$ vertices can be moved into $O(n)$ positions, and the evaluation of each resulting solution requires $O(n^3)$ computations.*

In the following we will show how the complexity of this calculation can be reduced to $O(n^3)$. The idea behind our procedure is to evaluate the expected cost difference of each neighbor τ' from a "close" neighbor τ'' whose cost is already known.

We observe preliminary that each neighbor τ' can be obtained from τ through a sequence of neighbors in each of which vertex i_s is moved one position forward or backward. In case of a forward move ($s < t$), such a sequence is as follows:

$$\begin{aligned} \tau &= (i_0, i_1, \dots, i_{s-1}, i_s, i_{s+1}, \dots, i_t, \dots, i_{2n+1}); \\ (i_0, i_1, \dots, i_{s-1}, i_{s+1}, i_s, \dots, i_t, \dots, i_{2n+1}) &\in N(\tau); \\ &\dots \\ (i_0, i_1, \dots, i_{s-1}, i_{s+1}, \dots, i_s, i_t, \dots, i_{2n+1}) &\in N(\tau); \\ \tau' &= (i_0, i_1, \dots, i_{s-1}, i_{s+1}, \dots, i_t, i_s, \dots, i_{2n+1}). \end{aligned}$$

Of course, all intermediate solutions need to be evaluated since they are part of $N(\tau)$. As a result, we simply need to compute the expected cost difference between pairs of tours in which a single vertex has been moved one position forward or backward. Moreover, it is worth noting that moving vertex i_s one position forward is equivalent to moving vertex i_{s+1} one position

backward. Hence, in the following only case $t = s + 1$ (referred to as *1-move*) will be considered.

Seven cases have to be considered. Let i_q be the companion of i_s (i.e., $i_q = i_s + n$ if $i_s \leq n$ or $i_q = i_s - n$ otherwise). Similarly, let i_r be the companion of i_{s+1} (i.e., $i_r = i_{s+1} + n$ if $i_{s+1} \leq n$ or $i_r = i_{s+1} - n$ otherwise).

Proposition 3.2 *If $q \in \{s + 2, \dots, 2n\}$, $r \in \{1, \dots, s - 1\}$, the following arc probability updates are required:*

- $p_{\tau'}(i_r, i_s) = 0$;
- $p_{\tau'}(i_h, i_s) = p_{\tau}(i_h, i_s)(1 - p_{i_{s+1}})$ $h = r + 1, \dots, s - 1$
- **if** $(p_{i_{s+1}} < 1)$ **then** $p_{\tau'}(i_s, i_k) = p_{\tau}(i_s, i_k)/(1 - p_{i_{s+1}})$ **else** $p_{\tau'}(i_s, i_k) = p_{\tau}(i_{s+1}, i_k)p_{i_s}$ $k = s + 2, \dots, q - 1$
- **if** $(p_{i_{s+1}} < 1)$ **then** $p_{\tau'}(i_s, i_q) = p_{\tau}(i_s, i_q)/(1 - p_{i_{s+1}})$ **else** $p_{\tau'}(i_s, i_q) = p_{\tau}(i_{s+1}, i_q)$
- **if** $(p_{i_s} < 1)$ **then** $p_{\tau'}(i_r, i_{s+1}) = p_{\tau}(i_r, i_{s+1})/(1 - p_{i_s})$ **else** $p_{\tau'}(i_r, i_{s+1}) = p_{\tau}(i_r, i_s)$
- **if** $(p_{i_s} < 1)$ **then** $p_{\tau'}(i_h, i_{s+1}) = p_{\tau}(i_h, i_{s+1})/(1 - p_{i_s})$ **else** $p_{\tau'}(i_h, i_{s+1}) = p_{\tau}(i_h, i_s)p_{i_{s+1}}$ $h = r + 1, \dots, s - 1$
- $p_{\tau'}(i_{s+1}, i_q) = 0$;
- $p_{\tau'}(i_{s+1}, i_k) = p_{\tau}(i_{s+1}, i_k)(1 - p_{i_s})$ $k = s + 2, \dots, q - 1$

Proof. When a *1-move* is performed, some arcs entering or leaving i_s and i_{s+1} change their probability while the remaining arc probabilities remain the same and do not need to be recomputed. In particular, the probabilities to be updated are:

$$\begin{aligned}
 & p_{\tau'}(i_h, i_s) \quad (h = 0, \dots, s - 1, s + 1); \\
 & p_{\tau'}(i_s, i_h) \quad (h = s + 1, \dots, n + 1); \\
 & p_{\tau'}(i_h, i_{s+1}) \quad (h = 0, \dots, s - 1); \\
 & p_{\tau'}(i_{s+1}, i_h) \quad (h = s + 2, \dots, n + 1).
 \end{aligned}$$

Probability $p_{\tau'}(i_h, i_s)$ remains equal to zero for $h = 1, \dots, r-1$ since the chain $i_{h+1}, \dots, i_{s-1}$ contains i_r but not i_{s+1} . For a similar reason, $p_{\tau'}(i_r, i_s)$ becomes equal to zero once i_s is moved one position ahead. As far as probability $p_{\tau'}(i_s, i_k)$ is concerned ($k = s+2, \dots, q-1$), two cases may occur: (i) if i_{s+1} is white, $p_{\tau}(i_s, i_k) = 0$ in which case $p_{\tau'}(i_s, i_k) = 0$ too, or the removal of i_{s+1} from chain $i_{s+2}, \dots, i_{k-1}$ reduce $p_{\tau'}(i_s, i_k)$ by a factor $(1 - p_{i_{s+1}})$; (ii) if i_{s+1} is black, then $p_{\tau'}(i_s, i_k) = p_{\tau}(i_{s+1}, i_k)p_{i_s}$ since chain $i_{s+2}, \dots, i_{k-1}$ does not change. Moreover, two cases have to be considered when calculating $p_{\tau'}(i_s, i_q)$: (i) if i_{s+1} is white, $p_{\tau}(i_s, i_q) = 0$ in which case $p_{\tau'}(i_s, i_q)$ is zero too, or the removal of i_{s+1} from chain $i_{s+2}, \dots, i_{q-1}$ reduce $p_{\tau'}(i_s, i_q)$ by a factor $(1 - p_{i_{s+1}})$; (ii) if i_{s+1} is black, then $p_{\tau'}(i_s, i_q) = p_{\tau}(i_{s+1}, i_q)$. Moreover, both $p_{\tau}(i_s, i_k)$ and $p_{\tau'}(i_s, i_k)$ are zero for $k > q$ since chain $i_{s+2}, \dots, i_{k-1}$ contains i_q . Using a similar reasoning, the probabilities associated to arcs entering or leaving i_{s+1} can be computed.

The proofs of the remaining six cases are similar to that of Proposition 3.2 and are omitted for the sake of brevity.

Proposition 3.3 *If $q \in \{1, \dots, s-1\}$, $r \in \{s+2, \dots, 2n\}$, the following are probability updates are required:*

- $p_{\tau'}(i_q, i_s) = p_{\tau}(i_q, i_s)(1 - p_{i_{s+1}})$
- $p_{\tau'}(i_q, i_{s+1}) = p_{\tau}(i_q, i_s)p_{i_{s+1}}$
- $p_{\tau'}(i_h, i_s) = p_{\tau}(i_h, i_s)(1 - p_{i_{s+1}}) \quad h = q+1, \dots, s-1$
- **if** $(p_{i_s} < 1)$ **then** $p_{\tau'}(i_h, i_{s+1}) = p_{\tau}(i_h, i_{s+1})/(1 - p_{i_s})$ **else** $p_{\tau'}(i_h, i_{s+1}) = p_{\tau}(i_h, i_s)p_{i_{s+1}} \quad h = q+1, \dots, s-1$
- **if** $(p_{i_{s+1}} < 1)$ **then** $p_{\tau'}(i_s, i_k) = p_{\tau}(i_s, i_k)/(1 - p_{i_{s+1}})$ **else** $p_{\tau'}(i_s, i_k) = p_{\tau}(i_{s+1}, i_k)p_{i_s} \quad k = s+2, \dots, r-1$
- $p_{\tau'}(i_{s+1}, i_k) = p_{\tau}(i_{s+1}, i_k)(1 - p_{i_s}) \quad k = s+2, \dots, r-1$
- $p_{\tau'}(i_s, i_r) = p_{\tau}(i_{s+1}, i_r)p_{i_s}$
- $p_{\tau'}(i_{s+1}, i_r) = p_{\tau}(i_{s+1}, i_r)(1 - p_{i_s})$

Proposition 3.4 *If $q \in \{1, \dots, s-1\}$, $r \in \{1, \dots, s-1\}$, $r < q$, the following arc probability updates are required:*

- $p_{r'}(i_q, i_s) = p_r(i_q, i_s)(1 - p_{i_{s+1}})$
- $p_{r'}(i_q, i_{s+1}) = p_r(i_q, i_s)p_{i_{s+1}}$
- $p_{r'}(i_h, i_s) = p_r(i_h, i_s)(1 - p_{i_{s+1}}) \quad h = q+1, \dots, s-1$
- *if $(p_{i_s} < 1)$ then $p_{r'}(i_h, i_{s+1}) = p_r(i_h, i_{s+1})/(1 - p_{i_s})$ else $p_{r'}(i_h, i_{s+1}) = p_r(i_h, i_s)p_{s+1} \quad h = q+1, \dots, s-1$*
- *if $(p_{i_{s+1}} < 1)$ then $p_{r'}(i_s, i_k) = p_r(i_s, i_k)/(1 - p_{i_{s+1}})$ else $p_{r'}(i_s, i_k) = p_r(i_{s+1}, i_k)p_s \quad k = s+2, \dots, 2n+1$*
- $p_{r'}(i_{s+1}, i_k) = p_r(i_{s+1}, i_k)(1 - p_{i_s}) \quad k = s+2, \dots, 2n+1$

Proposition 3.5 *If $q \in \{1, \dots, s-1\}$, $r \in \{1, \dots, s-1\}$, $r > q$, the following arc probability updates are required:*

- $p_{r'}(i_r, i_s) = 0$
- *if $(p_{i_s} < 1)$ then $p_{r'}(i_r, i_{s+1}) = p_r(i_r, i_{s+1})/(1 - p_{i_s})$ else $p_{r'}(i_r, i_{s+1}) = p_r(i_r, i_s)$*
- $p_{r'}(i_h, i_s) = p_r(i_h, i_s)(1 - p_{i_{s+1}}) \quad h = r+1, \dots, s-1$
- *if $(p_{i_s} < 1)$ then $p_{r'}(i_h, i_{s+1}) = p_r(i_h, i_{s+1})/(1 - p_{i_s})$ else $p_{r'}(i_h, i_{s+1}) = p_r(i_h, i_s)p_{s+1} \quad h = r+1, \dots, s-1$*
- *if $(p_{i_{s+1}} < 1)$ then $p_{r'}(i_s, i_k) = p_r(i_s, i_k)/(1 - p_{i_{s+1}})$ else $p_{r'}(i_s, i_k) = p_r(i_{s+1}, i_k)p_s \quad k = s+2, \dots, 2n+1$*
- $p_{r'}(i_{s+1}, i_k) = p_r(i_{s+1}, i_k)(1 - p_{i_s}) \quad k = s+2, \dots, 2n+1$

Proposition 3.6 *If $q \in \{s+2, \dots, 2n\}$, $r \in \{s+2, \dots, 2n\}$, $r < q$, the following arc probability updates are required:*

- $p_{r'}(i_h, i_s) = p_r(i_h, i_s)(1 - p_{i_{s+1}}) \quad h = 0, \dots, s-1$
- *if $(p_{i_s} < 1)$ then $p_{r'}(i_h, i_{s+1}) = p_r(i_h, i_{s+1})/(1 - p_{i_s})$ else $p_{r'}(i_h, i_{s+1}) = p_r(i_h, i_s)p_{s+1} \quad h = 0, \dots, s-1$*

- if $(p_{i_{s+1}} < 1)$ then $p_{\tau'}(i_s, i_k) = p_{\tau}(i_s, i_k)/(1 - p_{i_{s+1}})$ else $p_{\tau'}(i_s, i_k) = p_{\tau}(i_s, i_k)p_s$ $k = s + 2, \dots, r - 1$
- $p_{\tau'}(i_{s+1}, i_k) = p_{\tau}(i_{s+1}, i_k)(1 - p_{i_s})$ $k = s + 2, \dots, r - 1$
- $p_{\tau'}(i_s, i_r) = p_{\tau}(i_{s+1}, i_r)p_s$
- $p_{\tau'}(i_{s+1}, i_r) = p_{\tau}(i_{s+1}, i_r)(1 - p_s)$

Proposition 3.7 *If $q \in \{s + 2, \dots, 2n\}$, $r \in \{s + 2, \dots, 2n\}$, $r > q$, the following arc probability updates are required:*

- $p_{\tau'}(i_h, i_s) = p_{\tau}(i_h, i_s)(1 - p_{i_{s+1}})$ $h = 0, \dots, s - 1$
- if $(p_{i_s} < 1)$ then $p_{\tau'}(i_h, i_{s+1}) = p_{\tau}(i_h, i_{s+1})/(1 - p_{i_s})$ else $p_{\tau'}(i_h, i_{s+1}) = p_{\tau}(i_h, i_s)p_{s+1}$ $h = 0, \dots, s - 1$
- if $(p_{i_s} < 1)$ then $p_{\tau'}(i_s, i_k) = p_{\tau}(i_s, i_k)/(1 - p_{i_{s+1}})$ else $p_{\tau'}(i_s, i_k) = p_{\tau}(i_{s+1}, i_k)p_s$ $k = s + 2, \dots, q - 1$
- $p_{\tau'}(i_{s+1}, i_k) = p_{\tau}(i_{s+1}, i_k)(1 - p_{i_s})$ $k = s + 2, \dots, q - 1$
- if $(p_{i_{s+1}} < 1)$ then $p_{\tau'}(i_s, i_q) = p_{\tau}(i_s, i_q)/(1 - p_{i_{s+1}})$ else $p_{\tau'}(i_s, i_q) = p_{\tau}(i_{s+1}, i_q)$

Proposition 3.8 *If $i_{s+1} = i_s + n$ or $i_s = i_{s+1} + n$, the following arc probability updates are required:*

- $p_{\tau'}(i_h, i_{s+1}) = p_{\tau}(i_h, i_s)$ $h = 0, \dots, s - 1$
- $p_{\tau'}(i_s, i_k) = p_{\tau}(i_{s+1}, i_k)$ $k = s + 2, \dots, 2n + 1$

The computational complexity of the new neighborhood evaluation procedure is stated by the following proposition.

Proposition 3.9 *The previous neighborhood evaluation procedure takes $O(n^3)$ steps.*

Proof. *Each of the $O(n)$ vertices can be moved into $O(n)$ positions, and arc cost update requires $O(n)$ computations for each move.*

4 Heuristics

In order to assess the potential of our neighborhood evaluation procedure, we have implemented a local search and two Tabu Search (TS) procedures. The initial solutions for these algorithms have been provided by three constructive algorithms: a randomized heuristic, a nearest neighbor heuristic and an insertion algorithm in which requests are inserted in non increasing order with respect to probabilities. In the first TS procedure, the current solution is forced to remain feasible (*TS-feasible*). An iteration count t is maintained equal to the number of moves actually implemented. If a vertex is moved at iteration t , then operating on vertex is tabu until iteration $(t+q)$, where q is a parameter randomly selected in some interval $[q_{min}, q_{max}]$ [24]. After a tuning phase, these parameters have been set as follows: $q_{min} = 7$, $q_{max} = 12$. As is common in Tabu Search, a tabu move can be overridden if implementing it provides a better solution. In the second TS algorithm, condition (iii) is allowed to be violated (*TS-Infeasible*) in the spirit of Gendreau, Hertz and Laporte [9]. Let n_v be the number of violated precedence constraints. Then, the search process is guided by an artificial objective function $E[L_r] + \alpha n_v$. Penalty factor α is initially set equal to one tenth of the expected cost of the best known solution. Every h iterations, α is either halved or doubled. If the last h iterations were all feasible, then $\alpha := \alpha/2$. If they were all infeasible, then $\alpha := 2\alpha$. Otherwise α is unchanged. This strategy produces a mix of feasible and infeasible solutions and removes the need to calibrate α a priori.

5 Computational Results

The heuristics have been coded in C language and run on a PC with a Pentium III processor at 850 MHz. Test problems have been derived from a subset of the TSPLIB as follows. For each TSP instance, four PPDTSP instances have been generated by randomly choosing pickup and delivery vertices as well as request probabilities. Two versions of each procedure have been created: in the first one each neighbor of the current solution has been assessed from scratch while in the second one our neighborhood evaluation technique

has been used. Results are reported in Tables 1 to 8. Tables 1 and 2 show the objective function values and the computation times of the constructive heuristics, respectively. Table 3 and 4 illustrate the objective function values and the computation times of the local search heuristics, respectively. Tables 5 and 6 show the objective function values and the computation times of the TS-feasible heuristics, respectively. Finally, Tables 7 and 8 show the objective function values and the computation times of the TS-infeasible heuristics, respectively.

Each column heading is a combination of three of the following acronyms:

C: constructive heuristic;

L: local search procedure;

TSF: TS-feasible algorithm;

TSI: TS-infeasible algorithm;

R: randomized heuristic;

N: nearest neighbor heuristic;

I: insertion heuristic;

OBJ: expected cost;

DEV: percentage deviation of the expected cost from the best known solution;

SEC: computing time (in seconds) whenever formula (2) is used to assess solution neighbors;

SEC': computing time (in seconds) whenever our procedure is used to assess solution neighbors.

Computational results show that impressive savings in computation times are achieved when using our neighborhood evaluation procedure. Moreover, the best heuristics turn out to be the TS procedures albeit this comes at the expense of much larger computing times.

6 Conclusion

In this paper we have developed an efficient neighborhood search procedure. This method is two order of magnitude faster than the straightforward approach. Computational results show that, if embedded in a local search or Tabu Search framework, our approach results in a much faster heuristic.

Acknowledgement

This research was partially supported by the Italian National Research Council, by the Center of Excellence for High-Performance Computing, University of Calabria, Italy, and by the Canadian Natural Sciences and Engineering Research Council under grant OGP0039682. This support is gratefully acknowledged. Thanks are also due to Francesca Vocaturo for her help with code developing and for her insightful suggestions.

References

- [1] J. R. Birge, and F. V. Louveaux, *Introduction to Stochastic Programming*, Springer-Verlag, New York, 1997.
- [2] D.O. Casco, B.L. Golden BL, and E.A. Wasil, "Vehicle Routing with Backhauls: Models, Algorithms and Case Studies", *Vehicle Routing: Methods and Studies*, B. L. Golden, and A.A. Assad (Editors), North-Holland, Amsterdam, 1988, 121-147.
- [3] J.-F. Cordeau, and G. Laporte, A Tabu Search Heuristic for the Static Multi-Vehicle Dial-a-Ride Problem, *Transportation Research Part B*, 37 (2003), 579-594.
- [4] J.-F. Cordeau, and G. Laporte, The Dial-a-Ride Problem: Variants, Modeling Issues and Algorithms, *4OR - Quarterly Journal of the Belgian, French and Italian Operations Research Societies* 1 (2003), 89-101.
- [5] J. Desrosiers, Y. Dumas, and F. Soumis, A Dynamic Programming Solution to the Large-Scale Single-Vehicle Dial-a-Ride Problem with Time Windows, *American Journal of Mathematics and Management Science* 6 (1986), 301-325.
- [6] Y. Dumas, J. Desrosiers, and F. Soumis, The Pick-up and Delivery Problem with Time Windows, *European Journal of Operational Research* 54 (1991), 57-72.
- [7] M. Gendreau, A. Hertz, and G. Laporte, An Approximation Algorithm for the Traveling Salesman Problem with Backhauls, *Operations Research* 45 (1997), 639-641.

- [8] M. Gendreau, A. Hertz, and G. Laporte, The Traveling Salesman Problem with Backhauls, *Computers & Operations Research* 23 (1996), 501-508.
- [9] M. Gendreau, A. Hertz, and G. Laporte, A Tabu Search Heuristic for the Vehicle Routing Problem, *Management Science* 40 (1994), 1276-1290.
- [10] M. Goetschalckx, and C. Jacobs-Blecha, The Vehicle Routing Problem with Backhauls, *European Journal of Operational Research* 42 (1989), 39-51.
- [11] M. Gendreau, G. Laporte, and R. Seguin, Stochastic Vehicle Routing Problems, *European Journal of Operational Research* 88 (1996), 3-12.
- [12] P. Healy, and R. Moll, A New Extension of Local Search Applied to the Dial-a-Ride Problem, *European Journal of Operational Research* 83 (1995), 83-104.
- [13] B. Hunsaker, and M. Savelsberg, Efficient Feasibility Testing for Dial-a-Ride Problems, *Operations Research Letters* 30 (2002), 169-173.
- [14] P. Jaillet, A Priori Solution of the Traveling Salesman Problem in Which a Random Subset of the Customers are Visited, *Operations Research* 36 (1988), 929-936.
- [15] J. Jaw, A. R. Odoni, H. N. Psaraftis, and N.H.M. Wilson, A Heuristic Algorithm for the Multi-Vehicle Advance-Request Dial-a-Ride Problem with Time Windows, *Transportation Research B* 20 (1986), 243-257.
- [16] B. Kalantari, A.V. Hill, and S.R. Arora, An Algorithm for the Traveling Salesman Problem with Pickup and Delivery Customers, *European Journal of Operational Research* 22 (1985), 377-86.
- [17] G. Laporte, F.V. Louveaux, H. Mercure, A Priori Optimization of the Probabilistic Traveling Salesman Problem, *Operations Research* 42 (1994), 543-549.
- [18] A. Mingozzi, S. Giorgi, and R. Baldacci, An Exact Method for the Vehicle Routing Problem with Backhaul, *Transportation Science* 33 (1999), 315-329.

- [19] H. N. Psaraftis, A Dynamic Programming Solution to the Single Vehicle Many-to-Many Immediate Request Dial-a-Ride Problem, *Transportation Science* 14 (1980), 130-54.
- [20] H. N. Psaraftis, Analysis of an $O(N^3)$ Heuristic for the Single Vehicle Many-to-Many Euclidean Dial-a-Ride Problem, *Transportation Research* 17B (1983), 133-145.
- [21] H. N. Psaraftis, Scheduling Large-Scale Advance-Request Dial-a-Ride Systems, *American Journal of Mathematical and Management Sciences* 6 (1986), 327-367.
- [22] J. Renaud, F. F. Boctor, and G. Laporte, Perturbation Heuristics for the Pickup and Delivery Traveling Salesman Problem, *Computers & Operations Research* 29 (2002), 1129-1141.
- [23] J. Renaud, F.F. Boctor, and I. Ouenniche, A Heuristic for the Pickup and Delivery Traveling Salesman Problem, *Computers & Operations Research* 27 (2000), 905-916.
- [24] Y. Rochat, and E. D. Taillard, Probabilistic Diversification and Intensification in Local Search for Vehicle Routing, *Journal of Heuristics* 1 (1994), 147-167.
- [25] K. S. Ruland, and E.Y. Rodin, The Pickup and Delivery Problem: Faces and Branch-and-Cut Algorithm, *Computers & Mathematics with Applications* 33 (1987), 1-13.
- [26] M.W.P. Savelsbergh, and M. Sol, The General Pickup and Delivery Problem, *Transportation Science* 29 (1995), 17-29.
- [27] M.W.P. Savelsbergh, An Efficient Implementation of Local Search Algorithms for Constrained Routing Problems, *European Journal of Operational Research* 47 (1990), 75-85.
- [28] T.R. Sexton, and L.D. Bodin, Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times: I. Scheduling, *Transportation Science* 19 (1985), 378-410.

- [29] T.R. Sexton, and L. D. Bodin, Optimizing Single Vehicle Many-to-Many Operations with Desired Delivery Times: II. Routing, *Transportation Science* 19 (1985), 411-435.
- [30] T.R. Sexton, and Y. Choi, Pickup and Delivery of Partial Loads with Time Windows, *American Journal of Mathematical and Management Sciences* 6 (1986), 69-98.
- [31] R. Swihart, and J. D. Papastavrou, A Stochastic and Dynamic Model for the Single-Vehicle Pick-up and Delivery Problem, *European Journal of Operational Research* 114 (1999), 447-464.
- [32] L.J.J. Van der Bruggen, J.K. Lenstra, and P.C. Schuur, Variable-Depth Search for the Single-Vehicle Pickup and Delivery Problem with Time Windows *Transportation Science* 27 (1993), 298-311.

Table 1: Constructive Heuristic Objective Function Values and Deviations

INSTANCE	C-R-OBJ	C-R-DEV	C-N-OBJ	C-N-DEV	C-I-OBJ	C-I-DEV
gr17-1	1013.25	16.30	885.74	1.67	871.21	0.00
gr17-2	4414.67	99.87	2236.58	1.26	2208.76	0.00
gr17-3	1903.40	39.74	1409.30	3.47	1362.09	0.00
gr17-4	329.37	2.11	323.81	0.39	322.55	0.00
gr21-1	1991.35	24.27	1673.63	4.45	1602.40	0.00
gr21-2	5453.12	108.16	3513.42	34.11	2619.74	0.00
gr21-3	3705.54	47.12	2518.78	0.00	2575.04	2.23
gr21-4	1058.34	10.50	974.86	1.78	957.81	0.00
gr24-1	1149.34	29.69	996.69	12.47	886.20	0.00
gr24-2	2634.24	94.89	1595.39	18.03	1351.66	0.00
gr24-3	1988.81	60.60	1474.56	19.07	1238.35	0.00
gr24-4	926.61	26.64	781.51	6.81	731.68	0.00
fri26-1	663.35	0.96	657.04	0.00	664.79	1.18
fri26-2	880.27	1.27	869.26	0.00	903.91	3.99
fri26-3	817.85	0.00	821.70	0.47	855.30	4.58
fri26-4	577.96	0.82	578.49	0.91	573.26	0.00
swiss42-1	1897.12	75.30	1539.52	42.25	1082.23	0.00
swiss42-2	1590.40	56.27	1323.82	30.08	1017.71	0.00
swiss42-3	1532.93	51.49	1250.70	23.60	1011.90	0.00
swiss42-4	989.48	23.23	852.25	6.14	802.92	0.00
dantzig42-1	563.77	0.00	774.39	37.36	596.61	5.82
dantzig42-2	535.30	0.00	607.57	13.50	682.80	27.55
dantzig42-3	545.96	0.00	671.29	22.96	568.14	4.06
dantzig42-4	394.02	0.00	462.07	17.27	398.17	1.05
hk48-1	24153.08	151.59	11969.54	24.68	9600.22	0.00
hk48-2	26535.61	150.71	13581.95	28.32	10584.07	0.00
hk48-3	23470.92	155.93	12920.09	40.88	9170.68	0.00
hk48-4	9260.61	37.31	7928.21	17.55	6744.51	0.00
gr48-1	9722.13	136.41	5287.44	28.57	4112.36	0.00
gr48-2	10955.93	143.32	5698.08	26.55	4502.65	0.00
gr48-3	10350.95	127.37	5197.07	14.16	4552.53	0.00
gr48-4	4060.51	53.79	3518.49	33.26	2640.32	0.00
gr120-1	28029.95	306.55	8615.68	24.96	6894.64	0.00
gr120-2	25264.30	265.05	8018.86	15.87	6920.86	0.00
gr120-3	25692.99	270.43	8339.67	20.24	6936.02	0.00
gr120-4	13097.14	182.19	6241.60	34.48	4641.32	0.00
average deviation		81.94		20.18		4.36

Table 2: Constructive Heuristic Computation Times

INSTANCE	C-R-SEC	C-N-SEC	C-I-SEC
gr17-1	0.00	0.00	0.02
gr17-2	0.00	0.00	0.02
gr17-3	0.00	0.00	0.02
gr17-4	0.00	0.00	0.02
gr21-1	0.00	0.00	0.03
gr21-2	0.00	0.00	0.03
gr21-3	0.00	0.00	0.03
gr21-4	0.00	0.00	0.03
gr24-1	0.00	0.00	0.03
gr24-2	0.00	0.00	0.05
gr24-3	0.00	0.00	0.05
gr24-4	0.00	0.00	0.05
fri26-1	0.00	0.00	0.08
fri26-2	0.00	0.00	0.08
fri26-3	0.00	0.00	0.06
fri26-4	0.00	0.00	0.06
swiss42-1	0.00	0.00	1.08
swiss42-2	0.00	0.00	1.09
swiss42-3	0.00	0.00	1.11
swiss42-4	0.00	0.00	1.02
dantzig42-1	0.00	0.00	1.05
dantzig42-2	0.00	0.00	1.08
dantzig42-3	0.00	0.00	1.11
dantzig42-4	0.00	0.00	1.02
hk48-1	0.00	0.00	2.44
hk48-2	0.00	0.00	2.38
hk48-3	0.00	0.00	2.49
hk48-4	0.00	0.00	2.38
gr48-1	0.00	0.00	2.44
gr48-2	0.00	0.00	2.47
gr48-3	0.00	0.00	2.34
gr48-4	0.00	0.00	2.55
gr120-1	0.02	0.02	606.45
gr120-2	0.02	0.02	641.28
gr120-3	0.02	0.02	626.81
gr120-4	0.02	0.02	562.00
average time	0.01	0.01	68.48

Table 3: Local Search Objective Function Values and Deviations

INSTANCE	L-R-OBJ	L-C-DEV	L-N-OBJ	L-N-DEV	L-I-OBJ	L-I-DEV
gr17-1	871.21	0.00	871.21	0.00	871.21	0.00
gr17-2	2200.30	1.68	2163.91	0.00	2200.30	1.68
gr17-3	1375.94	1.33	1357.93	0.00	1357.93	0.00
gr17-4	321.90	0.00	322.46	0.17	322.25	0.11
gr21-1	1557.73	0.00	1557.73	0.00	1557.73	0.00
gr21-2	3228.12	23.22	2619.74	0.00	2619.74	0.00
gr21-3	2265.82	2.21	2216.76	0.00	2575.04	16.16
gr21-4	948.63	0.00	948.63	0.00	948.63	0.00
gr24-1	894.51	0.94	886.20	0.00	886.20	0.00
gr24-2	1519.70	19.08	1448.98	13.54	1276.19	0.00
gr24-3	1224.48	1.18	1301.28	7.53	1210.19	0.00
gr24-4	742.79	1.52	731.68	0.00	731.68	0.00
fri26-1	650.93	1.61	640.62	0.00	650.85	1.60
fri26-2	823.03	2.16	830.01	3.03	805.61	0.00
fri26-3	788.61	1.55	776.59	0.00	852.44	9.77
fri26-4	573.21	0.73	569.06	0.00	569.06	0.00
swiss42-1	1195.66	10.62	1164.88	7.77	1080.86	0.00
swiss42-2	1090.02	9.92	1015.46	2.40	991.63	0.00
swiss42-3	1118.91	12.84	1119.39	12.89	991.59	0.00
swiss42-4	761.46	0.00	801.26	5.23	802.92	5.45
dantzig42-1	555.57	1.05	591.19	7.53	549.79	0.00
dantzig42-2	533.17	0.00	545.07	2.23	636.07	19.30
dantzig42-3	540.11	0.00	560.93	3.86	560.93	3.86
dantzig42-4	391.33	0.00	391.33	0.00	391.33	0.00
hk48-1	10406.45	8.64	11301.11	17.99	9578.40	0.00
hk48-2	10018.75	0.00	12470.65	24.47	10271.07	2.52
hk48-3	10408.81	13.94	11382.37	24.60	9135.45	0.00
hk48-4	6793.38	4.10	6525.84	0.00	6533.90	0.12
gr48-1	4109.39	0.24	4586.45	11.87	4099.70	0.00
gr48-2	5173.55	15.13	4804.93	6.92	4493.81	0.00
gr48-3	4152.17	1.92	4750.03	16.60	4073.78	0.00
gr48-4	2638.62	0.06	2643.60	0.25	2636.95	0.00
gr120-1	7834.80	17.05	7415.21	10.79	6693.31	0.00
gr120-2	7158.65	8.02	7283.60	9.91	6627.17	0.00
gr120-3	7048.00	4.38	7381.35	9.32	6752.00	0.00
gr120-4	4877.31	5.47	4935.67	6.73	4624.48	0.00
average deviation		5.10		6.09		2.03

Table 4: Local Search Computation Times

INSTANCE	L-R-SEC	L-R-SEC'	L-N-SEC	L-N-SEC'	L-I-SEC	L-I-SEC'
gr17-1	0.08	0.00	0.03	0.00	0.03	0.00
gr17-2	0.06	0.02	0.02	0.02	0.02	0.02
gr17-3	0.05	0.02	0.02	0.00	0.02	0.00
gr17-4	0.08	0.02	0.02	0.00	0.02	0.00
gr21-1	0.23	0.05	0.08	0.02	0.11	0.00
gr21-2	0.23	0.03	0.16	0.02	0.03	0.00
gr21-3	0.23	0.03	0.09	0.02	0.03	0.00
gr21-4	0.25	0.03	0.14	0.02	0.13	0.02
gr24-1	0.45	0.05	0.23	0.03	0.06	0.02
gr24-2	0.33	0.03	0.17	0.02	0.14	0.03
gr24-3	0.55	0.08	0.22	0.03	0.13	0.02
gr24-4	0.45	0.06	0.28	0.03	0.06	0.00
fri26-1	0.27	0.03	0.14	0.02	0.23	0.03
fri26-2	0.27	0.03	0.17	0.03	0.33	0.03
fri26-3	0.30	0.05	0.14	0.02	0.11	0.02
fri26-4	0.36	0.06	0.17	0.02	0.14	0.02
swiss42-1	11.08	0.86	9.70	0.67	2.17	0.17
swiss42-2	9.58	0.77	5.08	0.37	4.84	0.39
swiss42-3	9.28	0.75	3.48	0.25	3.49	0.28
swiss42-4	10.81	0.83	14.19	1.03	1.31	0.09
dantzig42-1	3.33	0.30	4.25	0.30	4.52	0.39
dantzig42-2	1.92	0.17	3.83	0.30	4.30	0.55
dantzig42-3	2.28	0.20	4.70	0.36	2.89	0.27
dantzig42-4	4.19	0.36	7.27	0.59	2.28	0.20
hk48-1	39.84	2.83	13.97	0.95	4.67	0.33
hk48-2	43.24	2.98	16.30	1.08	6.39	0.44
hk48-3	32.28	2.34	18.70	1.25	5.41	0.39
hk48-4	32.33	2.25	40.55	2.48	15.19	0.98
gr48-1	34.86	2.36	11.91	0.77	5.30	0.36
gr48-2	34.59	2.44	7.05	0.47	5.20	0.36
gr48-3	37.78	2.59	11.69	0.77	9.45	0.63
gr48-4	31.39	2.09	26.88	1.61	3.95	0.27
gr120-1	14641.15	494.90	4723.30	137.86	2070.05	59.00
gr120-2	14577.39	477.19	3508.95	104.53	3086.39	96.87
gr120-3	13861.50	469.98	4558.30	137.58	1620.14	47.27
gr120-4	12569.94	408.22	5070.38	147.40	1169.45	32.16
average time	1555.38	52.08	501.74	15.02	223.03	6.71

Table 5: TS-Feasible Objective Function Values and Deviations

INSTANCE	TSF-R-OBJ	TSF-R-DEV	TSF-N-OBJ	TSF-N-DEV	TSF-F-OBJ	TSF-F-DEV
gr17-1	871.21	0.00	871.21	0.00	871.21	0.00
gr17-2	2200.30	1.68	2163.91	0.00	2200.30	1.68
gr17-3	1375.94	1.33	1357.93	0.00	1357.93	0.00
gr17-4	321.90	0.00	322.25	0.11	322.25	0.11
gr21-1	1557.73	0.00	1557.73	0.00	1557.73	0.00
gr21-2	2619.74	0.00	2619.74	0.00	2619.74	0.00
gr21-3	2265.82	2.21	2216.76	0.00	2575.04	16.16
gr21-4	948.63	0.00	948.63	0.00	948.63	0.00
gr24-1	894.51	0.94	886.20	0.00	886.20	0.00
gr24-2	1519.70	19.08	1444.93	13.22	1276.19	0.00
gr24-3	1224.48	1.18	1210.19	0.00	1210.19	0.00
gr24-4	742.79	1.52	731.68	0.00	731.68	0.00
fri26-1	650.93	1.61	640.62	0.00	650.85	1.60
fri26-2	823.03	2.16	830.01	3.03	805.61	0.00
fri26-3	788.61	1.55	776.59	0.00	852.44	9.77
fri26-4	569.06	0.00	569.06	0.00	569.06	0.00
swiss42-1	1195.66	10.62	1080.86	0.00	1080.86	0.00
swiss42-2	1038.28	4.71	1015.46	2.40	991.63	0.00
swiss42-3	1118.91	12.84	1119.39	12.89	991.59	0.00
swiss42-4	761.46	0.00	761.46	0.00	802.92	5.45
dantzig42-1	555.57	1.05	591.19	7.53	549.79	0.00
dantzig42-2	533.17	0.00	542.48	1.75	533.17	0.00
dantzig42-3	540.11	0.00	560.93	3.86	560.93	3.86
dantzig42-4	391.33	0.00	391.33	0.00	391.33	0.00
hk48-1	10406.45	8.64	11251.47	17.47	9578.40	0.00
hk48-2	10010.64	0.00	12470.65	24.57	10226.64	2.16
hk48-3	10408.81	13.94	11382.37	24.60	9135.45	0.00
hk48-4	6793.38	4.45	6504.07	0.00	653.48	0.46
gr48-1	4099.70	0.00	4586.45	11.87	4099.70	0.00
gr48-2	4693.39	4.44	4799.10	6.79	4493.81	0.00
gr48-3	4152.17	1.92	4664.96	14.51	4073.78	0.00
gr48-4	2638.62	0.06	2643.60	0.25	2636.95	0.00
gr120-1	7834.80	17.05	7415.21	10.79	6693.31	0.00
gr120-2	7155.52	8.33	7282.01	10.24	6605.41	0.00
gr120-3	7030.15	4.15	7381.35	9.35	6750.23	0.00
gr120-4	4877.31	5.47	4935.67	5.47	4624.48	0.00
average deviation		3.97		5.41		1.48

Table 6: TS-Feasible Computation Times

INSTANCE	TSF-R-SEC	TSF-R-SEC'	TSF-N-SEC	TSF-N-SEC'	TSF-F-SEC	TSF-F-SEC'
gr17-1	0.11	0.02	0.06	0.02	0.06	0.00
gr17-2	0.11	0.03	0.05	0.00	0.06	0.00
gr17-3	0.09	0.02	0.05	0.00	0.06	0.00
gr17-4	0.11	0.03	0.09	0.00	0.06	0.00
gr21-1	0.37	0.06	0.22	0.03	0.23	0.03
gr21-2	0.69	0.11	0.31	0.05	0.15	0.03
gr21-3	0.34	0.05	0.23	0.03	0.14	0.02
gr21-4	0.37	0.06	0.27	0.03	0.25	0.03
gr24-1	0.64	0.08	0.44	0.05	0.25	0.05
gr24-2	0.56	0.08	0.44	0.09	0.31	0.03
gr24-3	0.80	0.11	0.73	0.05	0.31	0.03
gr24-4	0.66	0.08	0.48	0.05	0.27	0.03
fri26-1	0.67	0.09	0.45	0.05	0.59	0.08
fri26-2	0.67	0.08	0.48	0.05	0.72	0.09
fri26-3	0.69	0.09	0.42	0.05	0.41	0.03
fri26-4	0.98	0.12	0.47	0.06	0.45	0.05
swiss42-1	17.49	1.30	25.38	1.78	9.09	0.69
swiss42-2	22.24	2.52	11.46	1.27	11.95	0.95
swiss42-3	15.77	1.27	9.78	0.70	10.98	0.91
swiss42-4	17.59	1.31	39.58	2.20	6.97	0.47
dantzig42-1	10.39	0.92	9.70	0.70	10.50	0.91
dantzig42-2	10.86	1.00	16.13	1.28	14.14	3.00
dantzig42-3	10.87	0.97	11.17	0.91	9.29	0.80
dantzig42-4	10.37	0.86	13.23	1.09	8.18	0.70
hk48-1	55.32	3.84	39.00	2.67	21.27	1.44
hk48-2	72.92	5.72	34.92	2.28	31.46	3.05
hk48-3	48.57	4.11	37.23	3.61	21.35	1.42
hk48-4	47.70	3.17	67.73	4.13	32.41	2.98
gr48-1	65.54	5.95	27.75	4.89	21.51	1.34
gr48-2	76.89	5.25	24.36	1.64	20.45	1.48
gr48-3	53.09	3.48	38.82	4.31	24.77	1.56
gr48-4	47.14	3.94	43.01	2.52	19.63	1.19
gr120-1	19386.40	644.56	9186.18	311.59	6531.42	343.76
gr120-2	20256.13	1514.84	9284.83	320.40	9093.09	287.48
gr120-3	24539.11	798.14	9064.51	271.14	8324.09	506.75
gr120-4	17417.42	657.70	9782.09	399.70	5653.72	165.16
average time	2283.05	101.72	1048.97	37.20	830.02	36.85

Table 7: TS-Infeasible Objective Function Values and Deviations

INSTANCE	TSI-R-OBJ	TSI-R-DEV	TSI-N-OBJ	TSI-N-DEV	TSI-I-OBJ	TSI-I-DEV
gr17-1	871.21	0.00	871.21	0.00	871.21	0.00
gr17-2	2200.30	1.68	2163.91	0.00	2200.30	1.68
gr17-3	1375.94	1.02	1409.30	3.47	1362.09	0.00
gr17-4	321.90	0.00	322.28	0.12	322.25	0.11
gr21-1	1557.73	0.00	1557.73	0.00	1563.70	0.38
gr21-2	2619.74	0.00	2628.20	0.32	2619.74	0.00
gr21-3	2216.76	0.00	2298.85	3.70	2575.04	16.16
gr21-4	948.63	0.00	948.63	0.00	948.63	0.00
gr24-1	925.24	4.41	907.34	2.39	886.20	0.00
gr24-2	1613.30	19.36	1444.93	6.90	1351.66	0.00
gr24-3	1210.19	0.00	1210.19	0.00	1210.19	0.00
gr24-4	731.68	0.00	731.96	0.04	731.68	0.00
fri26-1	650.93	1.61	640.62	0.00	650.85	1.60
fri26-2	823.03	2.16	830.01	3.03	805.61	0.00
fri26-3	788.61	1.55	776.59	0.00	852.44	9.77
fri26-4	569.06	0.00	569.06	0.00	569.06	0.00
swiss42-1	1080.86	0.00	1205.95	11.57	1080.86	0.00
swiss42-2	1038.28	5.51	1015.46	3.19	984.02	0.00
swiss42-3	1118.91	12.84	1087.48	9.67	991.59	0.00
swiss42-4	761.46	2.64	741.84	0.00	741.84	0.00
dantzig42-1	555.57	1.05	591.19	7.53	549.79	0.00
dantzig42-2	533.17	0.00	542.48	1.75	630.48	18.25
dantzig42-3	540.11	0.00	560.93	3.86	560.93	3.86
dantzig42-4	391.33	0.00	391.33	0.00	391.33	0.00
hk48-1	10269.26	7.21	11106.65	15.96	9578.40	0.00
hk48-2	9806.34	0.00	11762.98	19.95	10232.29	4.34
hk48-3	9931.30	8.71	10173.01	11.36	9135.45	0.00
hk48-4	6732.53	4.69	6731.85	4.68	6431.05	0.00
gr48-1	4099.70	0.00	4099.70	0.00	4099.70	0.00
gr48-2	4642.58	3.31	4799.10	6.79	4493.81	0.00
gr48-3	4121.28	1.67	4441.20	9.57	4053.46	0.00
gr48-4	2638.62	0.06	2641.86	0.19	2636.95	0.00
gr120-1	7254.68	8.39	6851.87	2.37	6693.31	0.00
gr120-2	6942.39	5.10	6644.15	0.59	6605.41	0.00
gr120-3	6881.33	6.02	6835.87	5.32	6490.71	0.00
gr120-4	4598.83	0.00	5051.92	9.85	4624.48	0.56
average deviation		2.95		4.19		1.75

Table 8: TS-Infeasible Computation Times

INSTANCE	TSI-R-SEC	TSI-R-SEC'	TSI-N-SEC	TSI-N-SEC'	TSI-I-SEC	TSI-I-SEC'
gr17-1	0.25	0.03	0.14	0.02	0.14	0.02
gr17-2	0.20	0.03	0.11	0.02	0.11	0.02
gr17-3	0.19	0.02	0.12	0.02	0.13	0.02
gr17-4	0.37	0.05	0.25	0.03	0.28	0.05
gr21-1	1.16	0.14	0.39	0.05	0.48	0.06
gr21-2	1.25	0.16	0.55	0.06	0.30	0.03
gr21-3	0.80	0.11	0.39	0.05	0.27	0.03
gr21-4	0.75	0.08	0.77	0.09	0.83	0.09
gr24-1	1.11	0.11	1.00	0.11	0.66	0.08
gr24-2	0.92	0.09	1.23	0.14	0.91	0.06
gr24-3	1.16	0.11	1.50	0.17	0.59	0.06
gr24-4	1.16	0.11	1.16	0.11	1.16	0.11
fri26-1	1.14	0.12	0.78	0.08	1.00	0.11
fri26-2	1.14	0.12	0.87	0.09	1.17	0.12
fri26-3	1.19	0.13	0.77	0.08	0.73	0.06
fri26-4	1.73	0.19	0.83	0.08	0.75	0.08
swiss42-1	39.50	2.41	26.56	1.64	16.30	1.00
swiss42-2	67.73	4.00	23.41	1.36	51.30	3.27
swiss42-3	30.08	1.92	24.39	1.44	30.34	1.98
swiss42-4	32.73	1.98	20.36	1.25	21.78	1.31
dantzig42-1	18.58	1.27	18.09	1.13	18.88	1.28
dantzig42-2	18.37	1.22	30.36	1.97	37.78	2.33
dantzig42-3	18.23	1.23	20.41	1.33	15.75	1.08
dantzig42-4	18.89	1.27	25.12	1.66	14.69	1.02
hk48-1	142.30	7.69	84.17	4.53	38.00	2.06
hk48-2	134.29	7.12	57.45	3.02	66.25	3.64
hk48-3	103.71	5.52	96.03	5.08	71.00	3.75
hk48-4	73.95	3.88	72.83	3.66	57.98	3.96
gr48-1	106.61	5.77	78.17	4.19	37.83	2.03
gr48-2	135.30	8.88	87.08	4.70	56.81	2.83
gr48-3	103.45	5.56	70.22	3.80	89.97	4.58
gr48-4	95.51	10.20	87.61	4.44	41.52	2.08
gr120-1	37963.18	822.43	31133.69	642.41	19732.39	407.40
gr120-2	53375.63	1154.91	25087.99	532.14	18975.68	410.02
gr120-3	45913.99	987.99	25493.05	775.77	33696.06	443.32
gr120-4	36307.55	769.71	27681.30	1003.58	16856.71	353.17
average time	4853.17	105.74	3061.92	83.34	2498.24	45.89